Microsoft

Azure Developer Community Call

Applied Serverless Architectures - Ask me Anything
January 12th, 2023

Christian Leinweber– Architect Cloud Apps
christian.leinweber@maibornwolff.de
christle

Dennis Zielke – Architect Cloud Apps
dzielke@microsoft.com
denniszielke

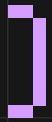
Code of Conduct

- Microsoft's mission is to empower every person and every organization on the planet to achieve more. This includes at this event, where we seek to create a respectful, friendly, and inclusive experience for all participants. As such, we do not tolerate harassing or disrespectful behavior, messages, images, or interactions by any event participant, in any form, at any aspect of the program including business and social activities, regardless of location.
- We do not tolerate any behavior that is degrading to any gender, race, sexual orientation or disability, or any behavior that would violate Microsoft's Anti-Harassment and Anti-Discrimination Policy, Equal Employment Opportunity Policy, or Standards of Business Conduct.
  In short, the entire experience must meet our culture standards.
- We encourage everyone to assist in creating a welcoming and safe environment. Please report any concerns, harassing behavior, suspicious or disruptive activity directly to us.
  Microsoft reserves the right to refuse admittance to or remove any person from this and/or future events at any time in its sole discretion.

© Microsoft Corporation.

Contents

- What are good serverless architectures?
- Successful architecture patterns
- Operational responsibilities
- Scaling factors and guidelines
- Cost implications
- Containers and Serverless
- Platform engineering



© Microsoft Corporation.

Who am i?

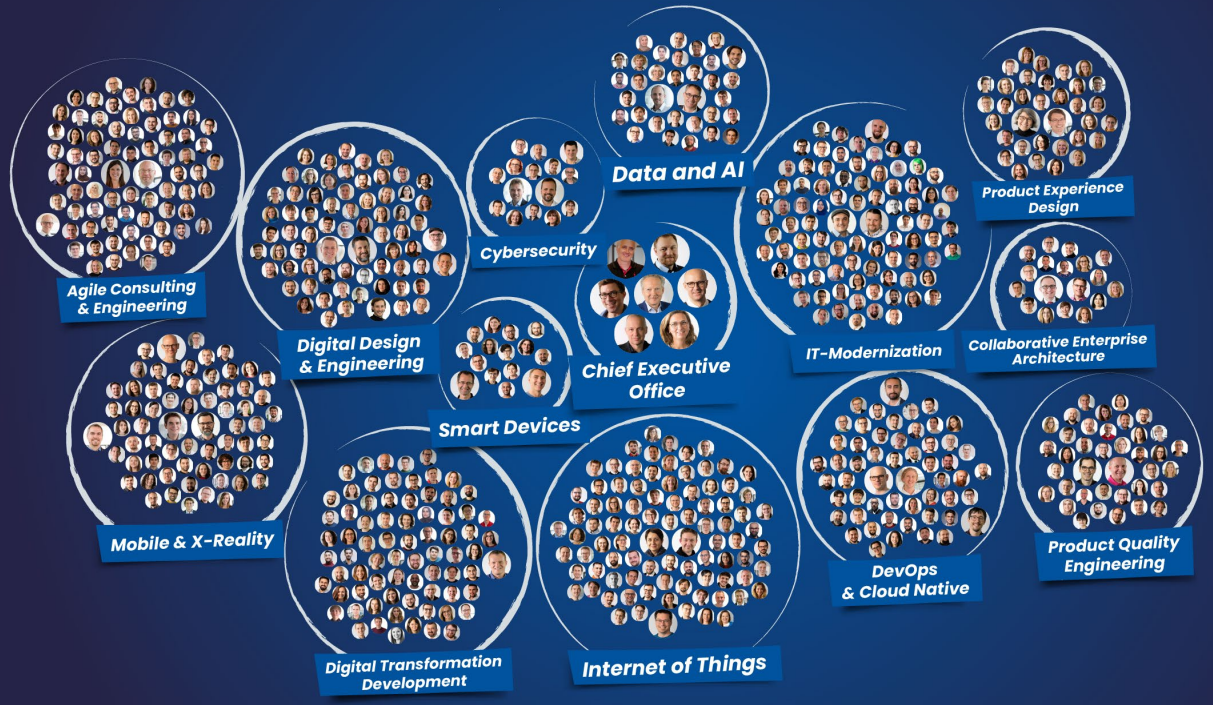
Christian Leinweber

- „Principal Architect“ @ MaibornWolff
- I live with my wife and my 3 kids near to Lüneburg
- I’m work at the Hamburg Office Location
- Me and my team build cloud-native applications
- We use serverless whereever it makes sense
- A lot of these projects focus on IoT systems

@christle_hh

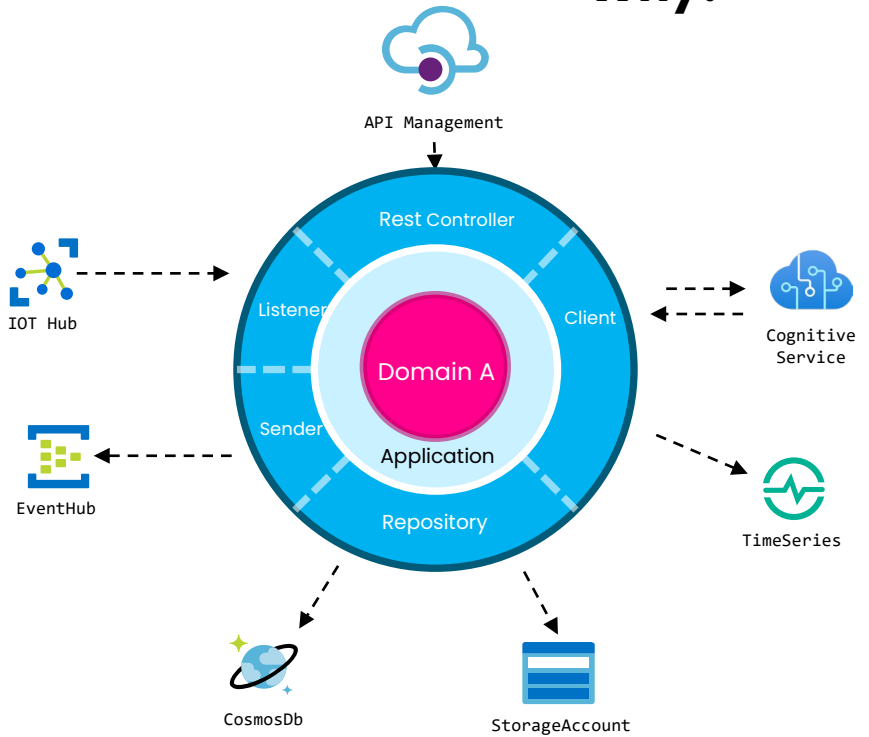
people. code. commitment.

- High investment in **Research & Development**
- More than 800 employees worldwide
- Locations: München, Augsburg, Darmstadt, Frankfurt, Berlin, Hamburg, Tunis, Valencia
- **12th consecutive year Great Place To Work**



With dedication to deliver and a passion for technology

A lot of applications make use of a hexagonal pattern. Why?

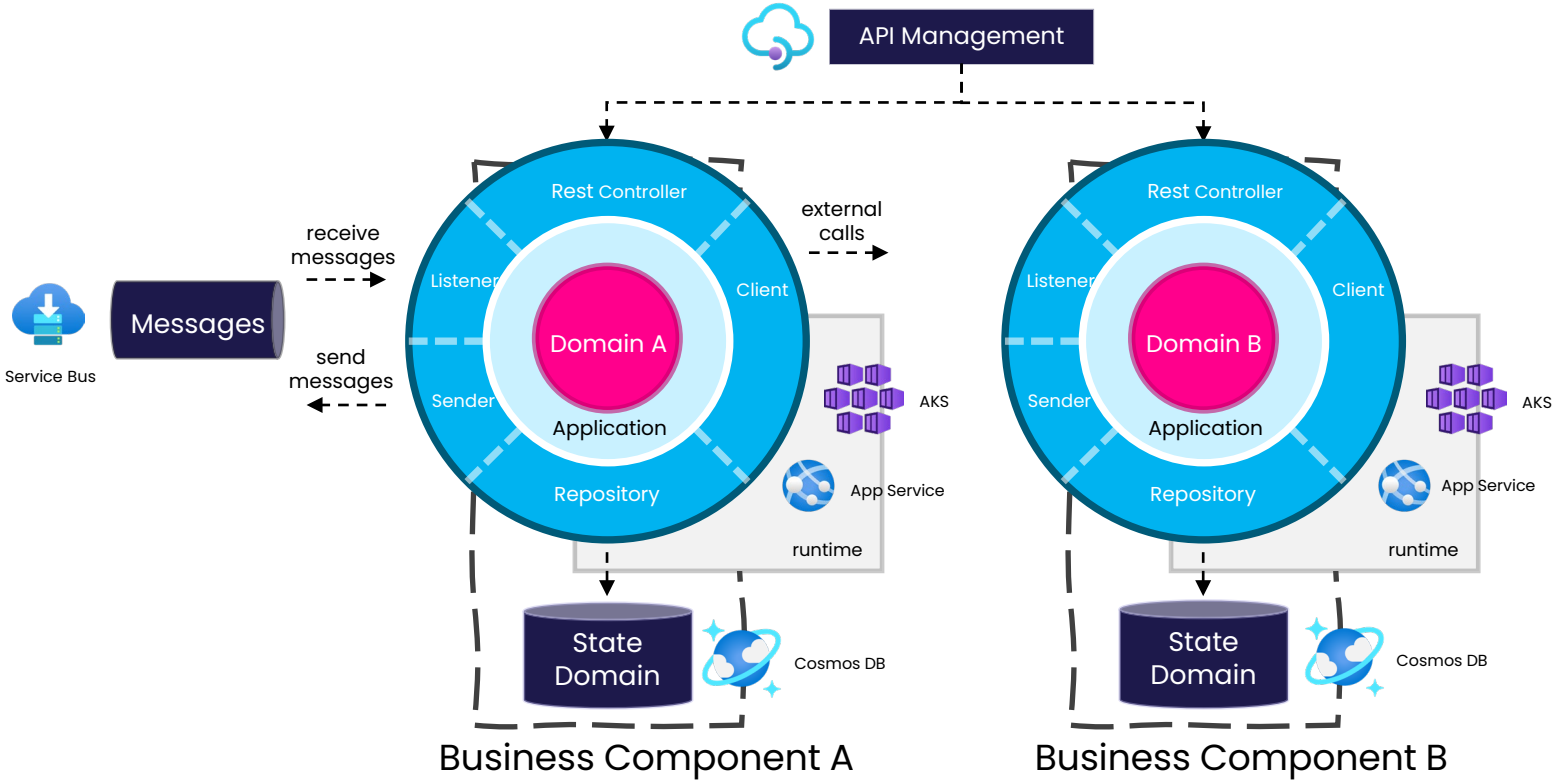


The **amount** of **external ressources** keep growing

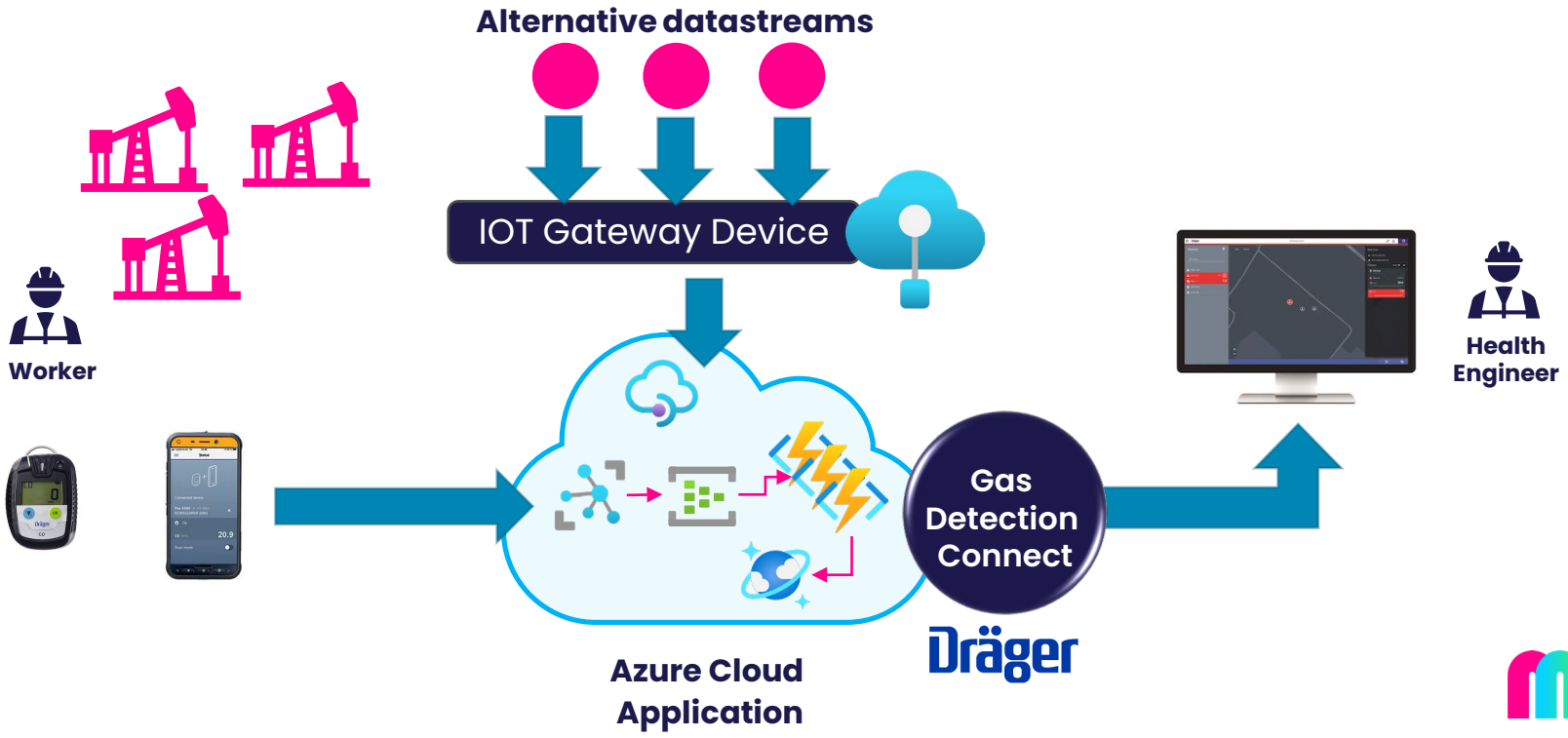
An **hexagonal architcecture** type support this **very well**

A very **domain orientated** approach

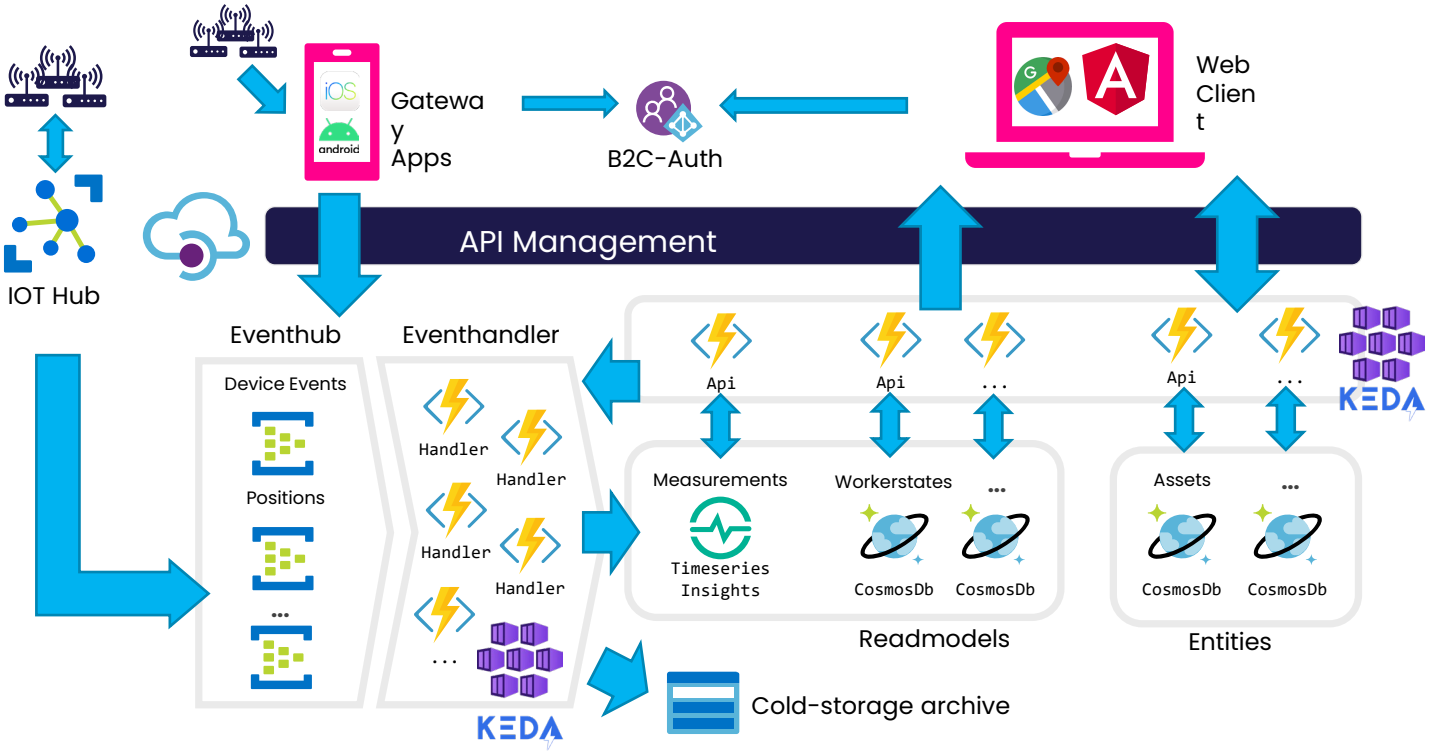
Hexagonal patterns work well with classic microservices



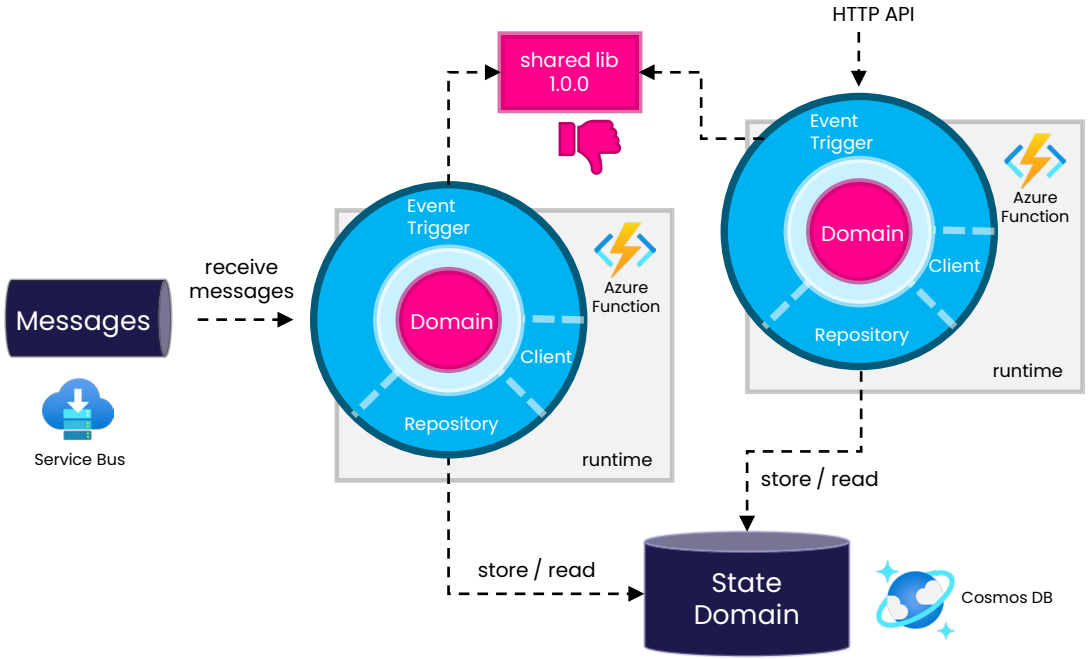
Current customer example based on serverless



Early solution draft based on Azure Functions



Using a Domain driven project setup in Azure Functions

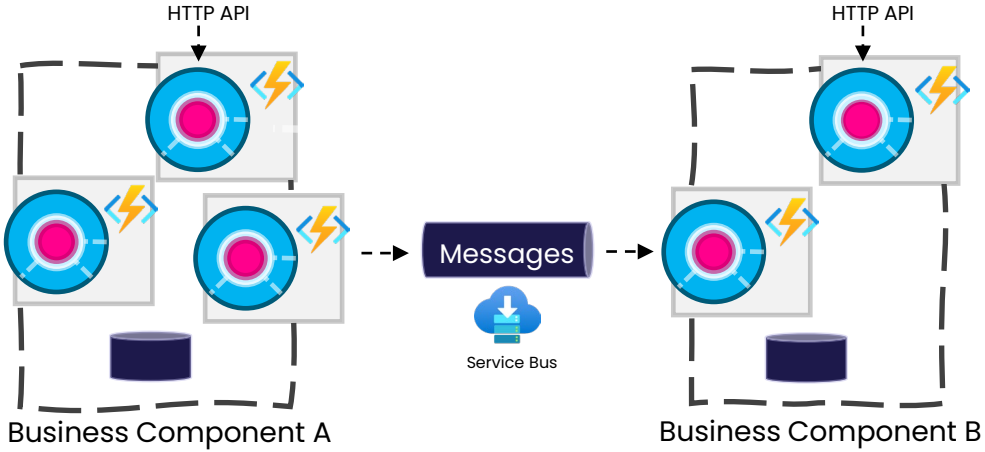


Technical motivation for an system cut and not a business perpective

A **shared versioned model lib** is even for microservices an **anti-pattern**

2 domain cores leading to **code duplications** and more **dependenies**

Use common microservice rules for virtual components

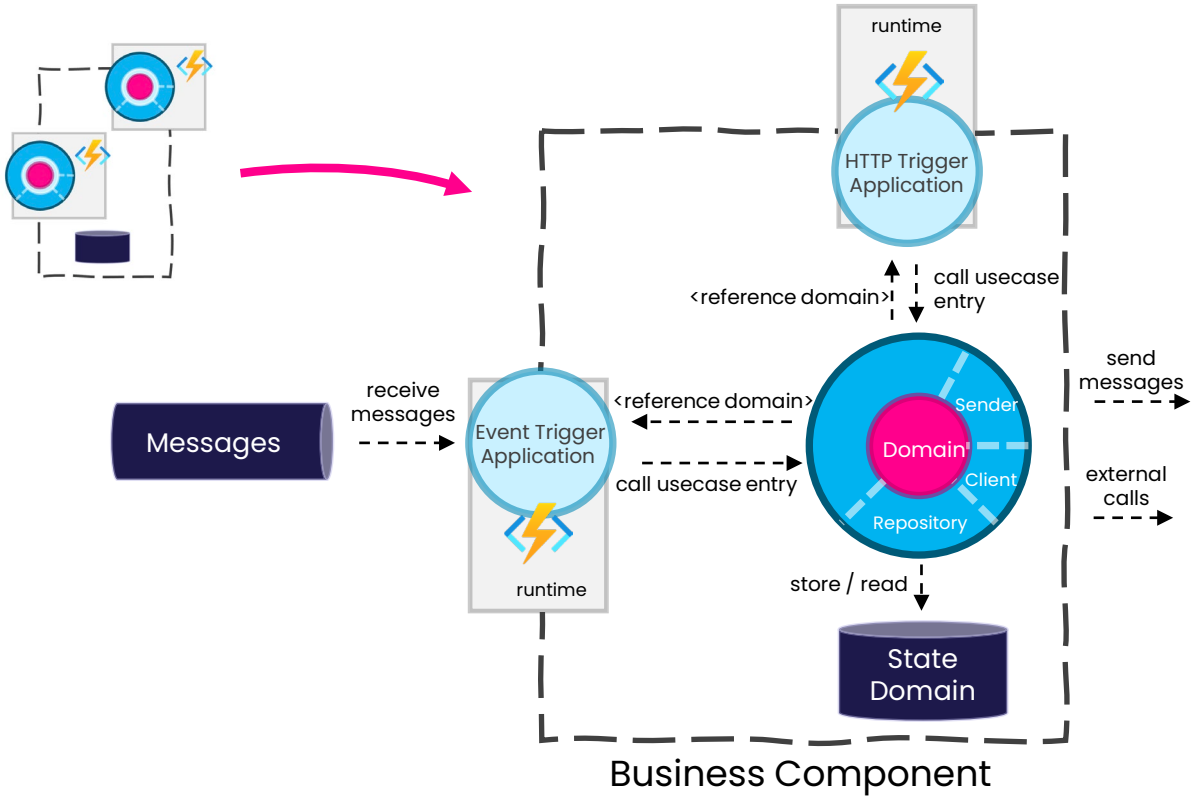


Only one datastore per virtual component

No cross componen t data access

Only **async communicatio n** between components

Design microservices with event triggered Applications



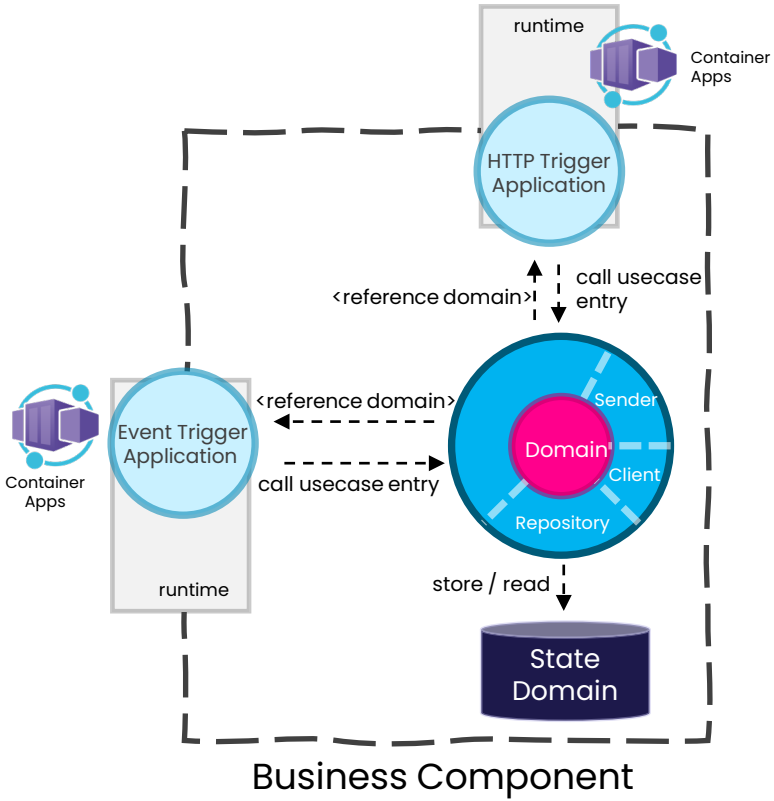
Build a **single domain core** for every business component as library

Split Domain core from **Applicatio n Layers**

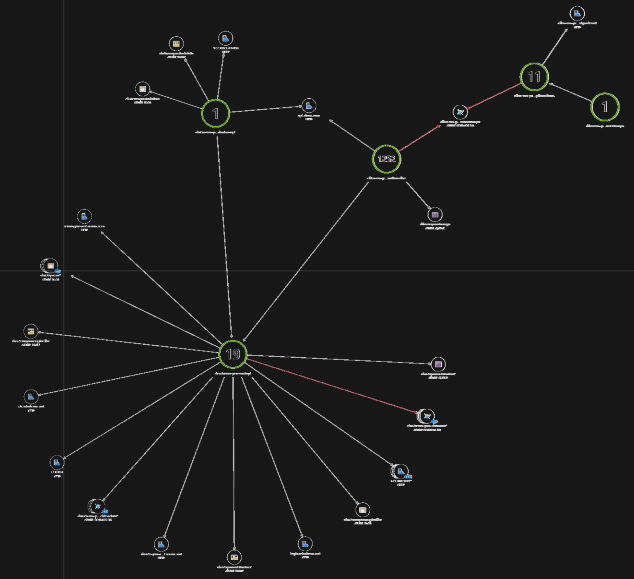
The Applicatio n Layer is **just an entrypoint**

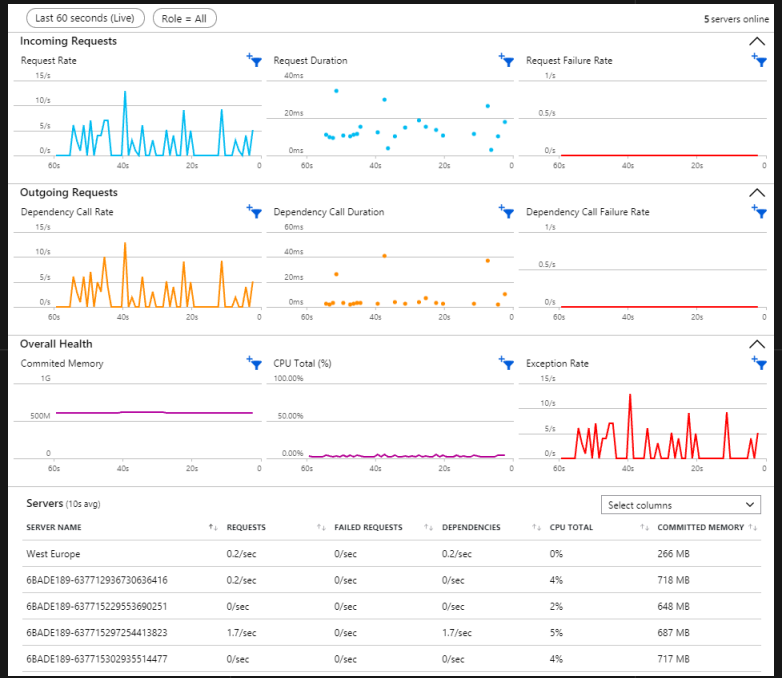
Scaling is still optimized **per Trigger Event**

It works very well with other serverless runtimes



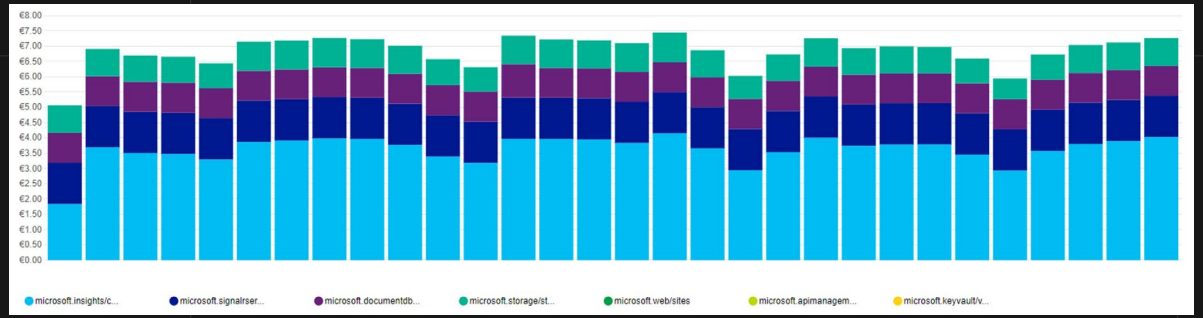
Observability

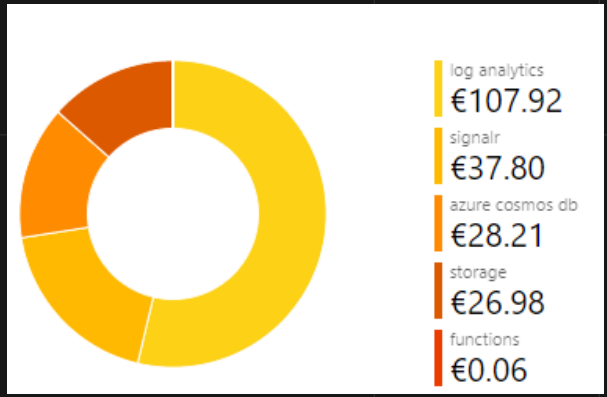




© Microsoft Corporation.

Cost management in serverless





© Microsoft Corporation.